

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 63-155264
(43)Date of publication of application : 28.06.1988

51)Int.Cl. G06F 15/347
G06F 9/44

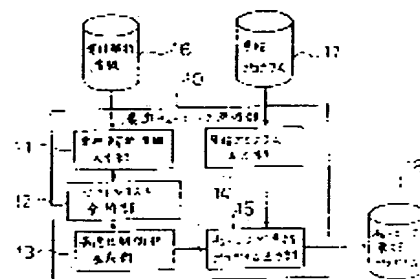
21)Application number : 61-302371 (71)Applicant : FUJITSU LTD
22)Date of filing : 18.12.1986 (72)Inventor : HIRABAYASHI TOSHIHIRO
MIURA SHINYA

54) LANGUAGE TUNING PROCESSING SYSTEM FOR VECTOR COMPUTER

57)Abstract:

PURPOSE: To remove performance deterioration factors due to vector processing by detecting automatically a loop which deteriorates in execution performance owing to the vector processing and incorporating an optimization control line which suppresses the vector processing of the loop.

CONSTITUTION: An execution analytic information input part 11 inputs execution analytic information 16 outputted by an execution analytic tool. A vector cost analytic part 12 calculates vector cost and scalar cost within a loop range of the vector processing by using the execution analytic information 16 inputted by the execution analytic information input part 11. An optimization control line generation part 13 generates the optimization control line for instructing the suppression of the vector processing to a compiler when the vector cost is larger than the scalar cost. The optimization control line is incorporated in a source program 17 inputted by a source program input part 14 and a tuning source program 18 is outputted through a tuning source program output part 15.



LEGAL STATUS

Date of request for examination]
Date of sending the examiner's decision of rejection]
Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
Date of final disposal for application]
Patent number]
Date of registration]
Number of appeal against examiner's decision of rejection]
Date of requesting appeal against examiner's decision of rejection]
Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

⑩ 日本国特許庁(JP)

⑪ 特許出願公開

⑫ 公開特許公報(A)

昭63-155264

⑬ Int.Cl.⁴

識別記号

庁内整理番号

⑭ 公開 昭和63年(1988)6月28日

G 06 F 15/347
9/44

3 2 2

G-8320-5B
A-8724-5B

審査請求 有 発明の数 1 (全5頁)

⑮ 発明の名称 ベクトル計算機用言語チューニング処理方式

⑯ 特 願 昭61-302371

⑰ 出 願 昭61(1986)12月18日

⑱ 発 明 者 平 林 俊 弘 神奈川県川崎市中原区上小田中1015番地 富士通株式会社
内

⑲ 発 明 者 三 浦 信 也 神奈川県川崎市中原区上小田中1015番地 富士通株式会社
内

⑳ 出 願 人 富士通株式会社 神奈川県川崎市中原区上小田中1015番地

㉑ 復 代 理 人 弁理士 小笠原 吉義

明 細 書

備えたことを特徴とするベクトル計算機用言語チューニング処理方式。

1. 発明の名称

ベクトル計算機用言語チューニング処理方式

3. 発明の詳細な説明

(概要)

2. 特許請求の範囲

ベクトル計算機を持つ処理装置上で動作させるプログラムの原始プログラムをチューニングするベクトル計算機用言語チューニング処理方式であって、

コンパイラによってベクトル演算命令が生成される原始プログラムについて、予めそのプログラムの実行解析情報に基づき、ベクトル化により実行性能が低下するループを自動検出し、そのループのベクトル化を抑制する最適化制御行を組み入れる手段を設けることにより、ベクトル化による性能低下要因を除去し、FORTRANプログラム等の原始プログラムをベクトル計算機向けに最適チューニングする。

チューニングするプログラムの実行解析情報に基づき、ベクトル化対象となるループ範囲について、ベクトル化による実行性能に関連するベクトルコストを分析するベクトルコスト分析手段(12)と、

(産業上の利用分野)

ベクトルコストがベクトル化しない場合におけるスカラコストより大きい場合に、チューニング対象原始プログラム中に、上記ループ範囲についてベクトル化を抑制することを指示する最適化制御行を組み入れる最適化制御行生成手段(13)とを

本発明は、FORTRANプログラム等の原始プログラムをベクトル計算機向けに最適チューニングするベクトル計算機用言語チューニング処理方式に関するものである。

データ処理装置による科学技術計算等では、大量のデータを高速に演算するベクトル計算機が用いられている。例えばFORTRAN言語等により記述された原始(ソース)プログラムを、コンパイラによって自動ベクトル化することが行われているが、その目的(オブジェクト)プログラムの実行性能に関する最適化が望まれる。

(従来の技術)

従来、FORTRANプログラム等の最適化を促進するために、プログラム中の各文について、実行回数や実行コスト等の実行解析情報を出力する実行解析ツールが用いられている。また、ベクトル演算命令を生成可能であるコンパイラでは、自動ベクトル化およびベクトル化された目的プログラムの最適化を促進するため各種オブティマイズ処理が行われている。

しかし、ベクトル化により性能低下を招く要因を検出して、それを除去する手段は、従来ないため、その最適化を行う場合には、人間が実行性能

を分析して、コンパイラに対し、いちいち最適化を指示する必要があった。

(発明が解決しようとする問題点)

例えばFORTRAN言語等により記述された原始プログラムをコンパイルして、ベクトル計算機用の目的プログラムを生成する場合、その性能を向上させるためには、より多くのDOLープを自動ベクトル化する必要があると考えられる。しかし、例えば短いベクトル長の演算の場合には、ベクトル命令によって処理するよりも、通常のスカラ命令によって処理したほうが実行速度が上がる場合がある。

コンパイラにおける自動ベクトル化において、ベクトルコストとスカラコストとの比較ができれば、コンパイラ内における最適化が可能であるが、例えば①DOLープのループ回数に変数である場合に、ベクトル長が不明であること、②動作するベクトル計算機の機種によってベクトルコストが変動することなどにより、コンパイラ内ではコス

ト比較ができない。

そのため従来、ベクトル化によって、かえって部分的に性能低下を招くことがあるという問題があった。

本発明は上記問題点の解決を図り、性能低下の要因となる短いベクトル長の演算を認識し、自動的に該当ループのベクトル化を抑止することによって、性能低下の要因を除去する手段を提供することを目的としている。

(問題点を解決するための手段)

第1図は本発明の原理ブロック図を示す。

第1図において、10は原始プログラムのベクトル化に関する最適チューニングを行う最適チューニング処理部、11は実行解析情報入力部、12はベクトルコストを分析するベクトルコスト分析部、13は最適化制御行を生成しベクトル化の抑止を指示する最適化制御行生成部、14は原始プログラムを入力する原始プログラム入力部、15はチューニングされた原始プログラムを出力す

るチューニング原始プログラム出力部、16はチューニング対象プログラムに関する実行解析情報、17はチューニング対象となる原始プログラム、18はチューニングされたチューニング原始プログラムを表す。

実行解析情報入力部11は、実行解析ツールの出力である実行解析情報16を入力する。この実行解析情報16は、実際のプログラムの走行または実行シミュレートによって、各文毎の実行回数等が出力されたものである。

ベクトルコスト分析部12は、実行解析情報入力部11が入力した実行解析情報16に基づき、ベクトル化対象となるループ範囲について、ベクトルコストとスカラコストとを計算する。

最適化制御行生成部13は、ベクトルコストがスカラコストよりも大きくなる場合に、ベクトル化を抑止することをコンパイラに指示する最適化制御行を生成する。そして、原始プログラム入力部14によって入力された原始プログラム17中に、その最適化制御行を組み入れ、チューニング

原始プログラム出力部15を介して、チューニング原始プログラム18を出力する。

(作用)

本発明によれば、ベクトルコスト分析部12により、実行解析情報に基づくコスト計算が行われ、ベクトル化されることによって性能低下を招く要因が自動検出される。そして、最適化制御行生成部13によって、最適化制御行が原始プログラム17中に組み入れられるので、コンパイル時には、その最適化制御行で指定されたループについてのベクトル化が抑止されることになる。

従って、コンパイラでは、ベクトル化により実際に性能が向上する部分だけ、ベクトル演算命令の生成を行い、ベクトル化によって性能が低下する部分については、通常のスカラ演算命令の生成を行うので、実行性能が向上することになる。

(実施例)

第2図は本発明が適用されるシステムの例、第

解析対象となるFORTRAN原始プログラム24中における制御移行に関連する部分に、実行回数をカウントする命令を埋め込むことなどにより、ループの繰返し回数やIF文の真率等の情報を含む実行解析情報16を出力する。なお、このFORTRAN実行解析部21は、従来からいわゆるFORTRANチューニングツールとして用いられているものを利用できる。

最適チューニング処理部10は、ベクトルコスト分析部12および最適化制御行生成部13によって、実行解析情報16に基づくコスト計算を行い、チューニング対象となるFORTRAN原始プログラム24中に、最適化制御行を組み込んだチューニングFORTRAN原始プログラム25を出力する。

FORTRANコンパイラ22は、チューニングFORTRAN原始プログラム25を機械語に翻訳するにあたって、最適化制御行判定部23により、ベクトル化を抑止する最適化制御行を検出すると、その指定範囲についてのベクトル化を抑

止する。3図は本発明の一実施例に係るベクトルコスト分析を説明するための図、第4図は本発明の一実施例に係る最適化制御行の生成を説明するための図、第5図は本発明の一実施例処理説明図、第6図は最適化による性能比較説明図を示す。

本発明は、例えば第2図に示すようなFORTRANのコンパイルを行う処理システムに適用される。

第2図において、第1図と同符号のものは第1図に示すものに対応し、20はCPUおよびメモリなどからなる処理装置、21はFORTRANプログラムの実行解析ツールであるFORTRAN実行解析部、22はFORTRAN言語により記述されたプログラムを計算機の機械語命令等からなる目的プログラムに翻訳するFORTRANコンパイラ、23は最適化制御行の有無を判定する最適化制御行判定部、24はFORTRAN原始プログラム、25はチューニングFORTRAN原始プログラム、26は目的プログラムを表す。

FORTRAN実行解析部21は、例えば実行

止する。

次に、第3図に従って、ベクトルコスト分析部12によるベクトルコストの分析例について説明する。

チューニング対象となるFORTRAN原始プログラムの各文のスカラコストがS_iで、その実行回数が第3図に示すようになっていたとする。例えば、ループ内の文1の実行回数は、ループ回数が100回であり、このDOループが10回実行されるので、1000回となる。この文1のループ当たりの平均実行回数 e_x は、100回となる。また、この例におけるIF文の真率、即ち、IF条件が成立する確率が5%であるとする、文2の実行回数は50回であり、そのループ当たりの平均実行回数 e_x は5回となる。

このDOループについてベクトル化する場合、ベクトル長はループ回数に対応し、例えばIF文に続く部分については、いわゆるマスク演算によって処理される。

従って、ベクトルコストとスカラコストとの比

較を行う場合には、マスク演算によるベクトル長が正しく反映されるように、スカラコストを補正することが必要となる。

以上の考慮により、D O ループのベクトルコスト V-COST は、次の式で求められる。

$$V-COST = \sum_i S_i * \frac{e x_i}{e x_i} * \alpha_i$$

$$\alpha_i = f(e x_i, c p u)$$

ここで、 S_i は各文のスカラコスト、

$e x_i$ は各文のループ当たりの平均実行回数、

$e x_i$ はループ先頭の文のベクトル長、

$c p u$ はベクトル計算機性能、

α_i は $e x_i$ および $c p u$ により求まるベクトル対スカラ性能比率である。

この α_i は、予めベクトル計算機による標準プログラムについての実測値に基づいて定められ、例えばテーブル化されて保持される。

こうして求められたベクトルコスト V と、ループ内のスカラコスト S_i の和との比較により、性能比較が行われる。

を計算する。

③ 次に、第3図で説明したD O ループのベクトルコスト V-COST を、実行解析情報と所定のベクトル対スカラ性能比率 α_i とに基づき計算する。

④ S-COST と V-COST との大小を比較し、V-COST が S-COST より大きければ、次の処理⑤を実行する。なお、等しい場合には、どちらでもよい。

⑤ 第4図に示すようにベクトル化抑止の最適化制御行を生成し、原始プログラム中に組み入れる。

第6図は、本発明の効果を説明するための性能比較を示す図である。第6図において、L1はスカラループ、L2はベクトル化によって性能が低下するループ、L3はベクトル化によって性能が向上するループを表している。

今、プログラムのベクトル化率が95%であると仮定する。また、ベクトル化による性能向上比が10倍、ベクトル化による性能低下比が5倍、

スカラコストが小さい場合には、第4図に示すように、最適化制御行の生成が行われる。第4図において、30は最適化制御行、31はベクトル化抑止範囲を表す。

即ち、FORTRAN原始プログラム24のD O 文の前に、例えば「*VOCL LOOP, SCALAR」という最適化制御行30が組み入れられ、チューニングFORTRAN原始プログラム25が生成される。コンパイラでは、この最適化制御行30を検出すると、これに続くD O ループをベクトル化抑止範囲31として認識し、この部分についてはベクトル演算命令の出力を抑止する。

第5図は、本発明の主要部分についての処理例を示している。以下、第5図に示す処理①～③に従って説明する。

① チューニング対象の原始プログラムについてベクトル化可能なD O ループを検出する。

② D O ループを検出したならば、実行解析情報に基づきループ内のスカラコストの和 S-COST

低下部分(L2の部分)がベクトル化可能部分の10%であるとする。第6図(イ)に示すオリジナルの実行時間、即ち、すべてスカラの演算命令により処理した場合の実行時間を100とすると、第6図(ロ)に示すベクトル化による実行時間T1は、次のようになる。

$$T1 = 95 \times 0.9 / 10 + 95 \times 0.1 \times 5 + 5$$

$$= 61.05$$

この実行時間T1は、従来方式によるベクトル化の性能と考えてよい。

一方、本発明により最適チューニングを行えば、その実行時間T2は、第6図(ハ)に示すようになり、次のようになる。

$$T2 = 95 \times 0.9 / 10 + 95 \times 0.1 + 5$$

$$= 23.05$$

相対性能比およびオリジナル比は、それぞれ次のようになる。

$$\text{相対性能比} = T1 / T2 = 61.05 / 23.05 = 2.64(\text{倍})$$

$$\text{オリジナル比} = 100 / 23.05 = 4.33(\text{倍})$$

(発明の効果)

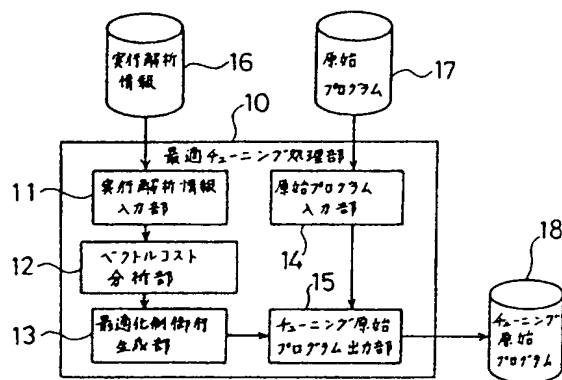
以上説明したように、本発明によれば、ベクトル化による性能低下要因が除去されるので、ベクトル計算機向けの最適チューニングが可能になる。

4. 図面の簡単な説明

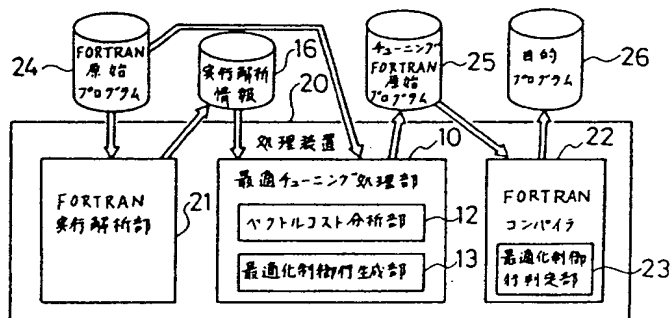
第1図は本発明の原理ブロック図、第2図は本発明が適用されるシステムの例、第3図は本発明の一実施例に係るベクトルコスト分析を説明するための図、第4図は本発明の一実施例に係る最適化制御行の生成を説明するための図、第5図は本発明の一実施例処理説明図、第6図は最適化による性能比較説明図を示す。

図中、10は最適チューニング処理部、11は実行解析情報入力部、12はベクトルコスト分析部、13は最適化制御行生成部、14は原始プログラム入力部、15はチューニング原始プログラム出力部を表す。

特許出願人 富士通株式会社
復代理人 弁理士 小笠原吉義



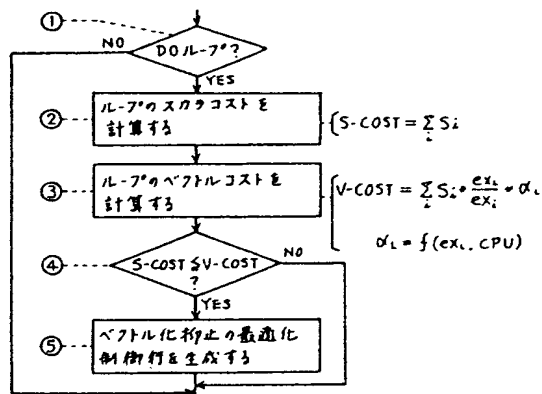
本発明の原理ブロック図
第1図



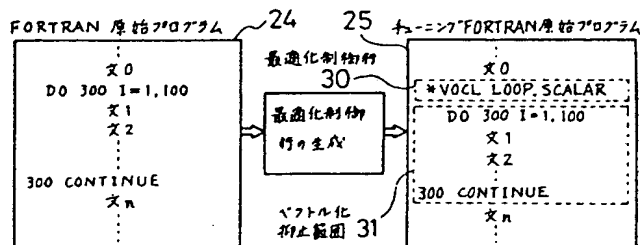
本発明の適用システム例
第2図

FORTRAN 原始プログラム	スカラコスト	実行回数	ex_i
文0	S_0	10回	10回
DO 300 I=1,100	S_1	1000回 (100回ループ)	100回
文1	S_2	1000回	100回
文2	S_3	1000回	100回
IF (T(I)) THEN	S_4	50回	5回
文2	S_5	50回 (真率)	5回
文m	S_m	50回 (5%)	5回
ENDIF			
300 CONTINUE		1000回	100回
文n	S_n	10回	10回

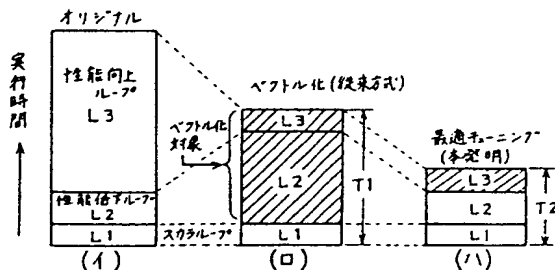
ベクトルコスト分析説明図
第3図



本発明の一実施例処理説明図
第5図



最適化制御行生成説明図
第4図



性能比較説明図
第6図